PROGETTO PROGRAMMAZIONE

Il progetto è stato realizzato da Jinxiufu LIn, Claudio Jin, Lorenzo Giordani, Limiao Jin per il corso di Programmazione (A.A. 2021-2022) della laurea triennale in Informatica dell’Università di Bologna.

## DESCRIZIONE DEL GIOCO

È un gioco di RPG dungeon. Il player dovrà esplorare le mappe e cercare di sopravvivere fra vari nemici accumulando i punti. La difficoltà salirà di livello in livello: aumenterà il numero dei nemici e la loro forza; inoltre aumenterà anche il numero degli artefatti che aiutano il player a sopravvivere . Il punteggio si aumenta sconfiggendo i nemici o raccogliendo gli artefatti. Il gioco termina se la vita del player arriva allo 0 oppure schiacciando il tasto ‘q’.

## IMPLEMENTAZIONE

L’idea alla base è la definizione di diversi oggetti (le classi Entity, Monster e Player) i quali interagiscono tra di loro tramite delle funzioni e delle strutture (Struct, Level, Graphics Physics).

### Classi:

#### Entity:

È la classe padre che contiene le informazioni fondamentali di entità presenti sulla mappa:

* Position: indica le coordinate x,y.
* texture: il simbolo che rappresenta l’entità sulla mappa.
* Active: indica lo stato di attività.

I suoi metodi, tranne quelli che restituiscono e modificano gli attributi, sono:

* il costruttore che inizializza gli attributi
* is\_move\_ok() : stabilisce se la posizione dell’entità è valida.
* death() : rende l’entità inattiva.

#### Monster:

È la classe figlio di classe Entity con attributi aggiuntivi:

* newposition: indica le coordinate x,y della posizione successiva.
* life: indica la vita dell’entità.

I nuovi metodi sono:

* damaged() : decrementa la vita fino a chiamare il metodo death().

#### Player:

È la classe figlio di classe Monster con attributi aggiuntivi:

* score: indica il punteggio.
* bullet: struttura che rappresenta il proiettile del player.

I metodi:

* scored() : aumenta il punteggio quando sconfigge i nemici.
* lifeUp() : aumenta la vita del player.

## INTERAZIONI:

### Level

Definisce la struttura Level che è una lista bidirezionale che memorizza tutti i livelli, le entità (monster, artefatti,telep) e altri attributi di base come:

* num\_lev indica il numero di livello.
* id\_map stabilisce quale mappa da disegnare.
* n\_monsters e n\_bonus indicano rispettivamente il numero dei mostri e degli artefatti presenti in ogni livello.
* EntrDoor e ExitDoor rappresentano le coordinate y delle porte di entrata ed uscita.
* tp\_flag e Show\_tp sono rispettivamente la flag per la presenza di teletrasporti e per mostrare questi poteri sulla mappa.
* DoorOpened indica se la porta di uscita è stata aperta o meno.
* next e prev sono i due puntatori al livello successivo e quello precedente.

La funzione newlv() inizializza un nuovo livello stabilendo gli attributi sopra indicati; invece la funzione Switchlv() cambia il livello corrente.

### Graphics

Definisce le funzioni necessarie per mostrare la mappa.

### Physics

Contiene le funzioni

* che regolano il movimento:
* controll() : controlla player
* updateMonsterPosition() : aggiorna la posizione dei nemici a random oppure avvicinandosi all player.
* che aggiornano le posizione delle entità:
* updateMonster(), updateBonus() e updatePlayer() aggiornano le posizioni
* CheckPostion(): controlla se la posizione iniziale sia valida altrimenti viene assegnata una nuova.
* CheckDoor(): stabilisce se tutti i nemici siano morti e quindi apre la porta d’uscita.
* che stabiliscono i contatti tre le varie entità:
* bullet\_hit() : funzione booleana che stabilisce se sono in contatto i nemici e il proiettile.
* Player\_hit() : funzione booleana che stabilisce se sono in contatto il player e le altre entità.
* Hitbox() : stabilisce gli effetti dei contatti tra le entità.

## SUDDIVISIONE DEL LAVORO

Il lavoro è stato eseguito tra i componenti del gruppo nel seguente modo:

* Lin
  + la base del lavoro e la coordinazione delle varie parti.
* Jin C.
  + Effetti dei contatti e le altre funzioni collegate
* Giordani
  + controlli e aggiornamenti delle varie entità e le classi
* Jin L.
  + Grafica e varie funzioni relative

La suddivisione è indicativa e le parti sono state svolte in collaborazioni con gli altri componenti del gruppo.

# 